3-2002

Efficient Algorithms for Wavelength Converter Placement

Ling Li
Iowa State University

Arun K. Somani
Iowa State University, arun@iastate.edu

Efficient Algorithms for Wavelength Converter Placement

Abstract

In this paper, we consider the optimal converter placement problem for a given number of converters on a path topology in an all-optical WDM networks. The placement of converters on a path divides the path into segments. A segment is defined as a set of links between two consecutive converters on a path. We first introduce and prove that optimal placement considering end-to-end performance is obtained when segments on a path have equal blocking probability. This result is then used to achieve optimal converter placement using both the link-load independence model and link-load correlation model. It is not always possible to divide a path into segments with equal blocking probability due to the arbitrary values of the load on each link. Three implementation algorithms that approximately achieve minimum blocking probability with linear complexity are then proposed. The algorithms can be readily extended to ring networks.

Disciplines

Digital Communications and Networking | Systems and Communications

Comments

This is a manuscript of an article published as Li, Ling, and Arun K. Somani, "Efficient Algorithms for Wavelength Converter Placement," *Optical Networks Magazine* 3, no. 2 (2002): 54-62. Posted with permission.

EFFICIENT ALGORITHMS FOR WAVELENGTH CONVERTER PLACEMENT IN ALL-OPTICAL NETWORKS *

Ling Li *Arun Somani*

Department of Electrical and Computer Engineering
Iowa State University
Ames, IA 50010
Phone: (515) 294-0941, Fax: (515) 294-8432
e-mail: {lingli, arun}@iastate.edu

ABSTRACT

In this paper, we consider the optimal converter placement problem for a given number of converters on a path topology in all-optical WDM networks. The placement of converters on a path divides the path into segments. A segment is defined as a set of links between two consecutive converters on a path. We first introduce and prove that optimal placement considering end-to-end performance is obtained when the segments on a path have equal blocking probability. This result is then used to achieve optimal converter placement using both the link-load independence model and link-load correlation model. It is not always possible to divide a path into segments with equal blocking probability due to the arbitrary values of the load on each link. Three implementation algorithms that approximately achieve minimum blocking probability with linear complexity are then proposed. The algorithms can be readily extended to ring networks.

1. INTRODUCTION

Wavelength-routed all-optical networks have emerged as the key to fulfil the bandwidth requirement and provide new services in this information age. In such networks, the wavelength continuity constraint, assigning the same wavelength to route a connection on every link on a path, is a well-known problem. To reduce the blocking probability, wavelength converters, which can change an incoming wavelength to another, are proposed to use [1]. The benefits of employing wavelength converters are discussed in [2, 3]. Since the cost of an all-optical wavelength converter is likely to remain high, sparse wavelength conversion and limited wavelength conversion are studied in [4, 5]. A network with only a few nodes having full conversion capability is called a network with sparse wavelength conversion. The results in [4] show placing converters on a fraction of nodes of a network is sufficient to ensure high network performance.

An interesting problem in sparse wavelength conversion networks is how to place a small number of converters so that the network performance is optimized. This problem was first addressed in [6]. A solution using dynamic programming was proposed to optimize the performance on a path, a bus network, and a ring network. However, the complexity of this solution is $O(N^2K)$ when the blocking probability of end to end calls is optimized on a path. Here N is the number of nodes on the path, and K is the number of converters being placed.

In this paper, we develop a new linear complexity algorithm to place converters on a path to minimize the blocking probability (maximize the success probability). Given the number of nodes, N, and the number of converters, K, the basic idea of our algorithm is that we divide the path into K+1 segments such that the blocking probability on each segment is equal. It is shown that the blocking probability is minimized if each segment has the same blocking probability. However, it is not always possible to divide a path into segments with equal blocking probabilities, due to the arbitrary values of the loads on each link. Three algorithms are proposed in this paper to divide a path into segments with approximately equal blocking probabilities. The results of using these algorithms are compared with the optimal solutions obtained by using dynamic programming method of [6].

This paper is organized as follows. The end-to-end optimal converter placement is studied using a link-load independence model in section 2. We first prove that the optimal placement is obtained if the blocking probability on each segment is equal. Three algorithms with linear complexities are proposed to obtain the segments with approximately equal blocking probabilities. A link-load correlation model is used for optimal converter placement in section 3. An analytical approximation is proposed and shown that the segmentation algorithms are also applicable when the link-load correlation model is used. We make some concluding remarks in section 4.

2. CONVERTER PLACEMENT WITH END-TO-END PERFORMANCE OPTIMIZATION

We consider a path of length N, as shown in Figure 1. Let the number of converters to be placed be K. Let the nodes along the path be numbered $0, 1, \ldots, N$, and let the link loads per wavelength on link i be $\rho_i, i = 0, \ldots, N-1$. We employ a binomial model [3] to compute the performance; that is, we assume that a wavelength on link i is occupied with probability ρ_i, and the occupancy is statistically independent of other wavelengths on the same link and on other links. The converter placement for minimizing the blocking probability for end-to-end calls (from node 0 to node N) is considered in this section. A segmentation idea is introduced and proved that optimal placement can be obtained by dividing a path into K+1 segments such that each segment has equal blocking probability. Three implementation algorithms of this idea are also proposed in this section.

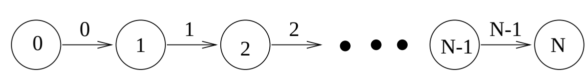

Figure 1. a path with N links

*THIS WORK WAS SUPPORTED BY THE NSF UNDER GRANTS NCR-9628165 AND NCR-9796318.

2.1. A segmentation method for converter placement on a path

We define a *segment* to be the set of links between two consecutive converters. Thus K converters divide the N link path into $K+1$ segments. Let $V = (0, v_1, \ldots, v_K, N)$ be the converter placement vector denoting the first node, the converter locations, and the last node of the path. v_i is the node with the ith converter. The ith segment is from node v_{i-1} to node v_i (the first segment is from node 0 to node v_1, and the $K+1$ segment is from node v_K to node N). Without any loss of generality, we count converter locations from the left to the right. Let $L(v_{i-1}, v_i)$ be a set of links on segment i from node v_{i-1} to node v_i.

When the wavelength utilization on each link is assumed equal and the link-load correlation is neglected, intuition suggests that the optimal solution is to place converters uniformly (each segment has the equal length) on the path. This placement has been proved optimal in [6]. We consider a more general scenario in which link loads may non-uniformly distributed. We assume in this section that link loads are independent. The effect of link-load correlation is studied in the next section. Let F be the number of wavelengths on each link. Let $S(V)$ be the success probability of an end-to-end call given the placement vector is V,

$$S(V) = \prod_{i=1}^{K+1} f_i \tag{1}$$

where f_i is the success probability on segment i and is given by

$$f_i = [1 - (1 - \prod_{j \in L(v_{i-1}, v_i)} \bar{\rho_j})^F] \tag{2}$$

and $\bar{\rho_j} = 1 - \rho_j$, is the success probability on link j.

Since we assume that wavelength utilization may or may not uniformly distributed over links, p_i may or may not equal to p_j if $i \neq j$. Our goal is to select a placement vector V such that $S(V)$ is maximized. Instead of dividing the path into equal length segments as in [6], we divide the path such that the success probability on each segment is the same. We show that the placement obtained using this idea is optimal by proving the following results:

Lemma 1 *Let $Z = (z_1, z_2, \ldots, z_{K+1})$ be a vector of $K+1$ real numbers z_i, $-\infty \leq z_i \leq 0$, $i = 1, 2, \ldots, K+1$, the function*

$$G(Z) = \prod_{i=1}^{K+1} (1 - (1 - e^{z_i})^F) \quad \textit{given} \quad \sum_{i=1}^{K+1} z_i = \textit{constant}$$

is maximized if

$$z_1 = z_2 = \ldots = z_{K+1}.$$

The proof of this lemma is shown in the appendix.

Theorem 1 *An optimal placement for end-to-end performance is achieved if the success probability on each segment is equal on the path.*

Proof: Let Y_i be the success probability on one wavelength on segment i. Then,

$$Y_i = \prod_{j \in L(v_{i-1}, v_i)} \bar{\rho_j}. \tag{3}$$

It is ready to see that $0 \leq Y_i \leq 1$. Since $\bar{\rho_j} = 1 - \rho_j$ is known on every link, $\prod_{i=1}^{K+1} Y_i = \prod_{j=0}^{N-1} (\bar{\rho_j})$ is a constant. $S(V)$ from Eqs. 1, 2 and 3 becomes

$$S(V) = \prod_{i=1}^{K+1} (1 - (1 - Y_i)^F). \tag{4}$$

The goal is to prove that $S(V)$ is maximized when $Y_1 = Y_2 = \ldots = Y_{K+1}$ given $\prod_{i=1}^{K+1} Y_i$ =constant. However, the global maximum value is hard to prove because the constraint of Y_i is not a convex set. We use the following substitution to convert the constraint to a convex set.

Let $Y_i = e^{z_i}, -\infty \leq z_i \leq 0$. Then $S(V)$ becomes

$$S(V) = \prod_{i=1}^{K+1} (1 - (1 - e^{z_i})^F) \tag{5}$$

From lemma 1 we know that the maximum value of $S(V)$ is obtained when $z_1 = z_2 = \ldots = z_{K+1}$, that is, $Y_1 = Y_2 = \ldots = Y_{K+1}$. ■

Theorem 1 provides insight into the converter placement problem and helps us solve the problem without computing the blocking probability for every s-d pair. Let the wavelength utilization on all the links (a set of positive real numbers between 0 and 1) be represented by an indexed set. The optimal solution is obtained by dividing the set into K+1 subsets such that each subset consists of consecutive elements in the set and the product of elements in each subset are all equal.

2.2. Implementation of the segmentation method

To divide an N link path into $K+1$ segments such that each segment has the equal success probability ($f_1 = f_2 = \ldots = f_{K+1}$) is not trivial. One of the difficulties is that the success probabilities on links, $\bar{\rho}, 0 \leq j \leq N-1$, may vary significantly. From Eq. 2, we know that

$$f_i = f_j \quad \text{if and only if } Y_i = Y_j \ .$$

A key observation here is that the geometric mean of the success probability on one wavelength on each segment can be easily obtained. To identify a segment, we compute the success probability on one wavelength of successive links and compare it with the geometric mean of the success probability. The success probabilities on all the segmentations are approximately equal if all of them are approximately equal to the geometric mean of the success probability.

Let M be the geometric mean of the success probability on each segment for one wavelength. Then,

$$M = \sqrt[K+1]{\prod_{i=1}^{K+1} Y_i} = \sqrt[K+1]{\prod_{j=0}^{N-1} \bar{\rho_j}} \ . \tag{6}$$

Recall that v_{i-1} is the placement of the $(i-1)$th converter. The next converter placement, v_i, is obtained by selecting consecutive links after node v_{i-1} on the path such that the product, $f_i' = \prod_{j \in L(v_{i-1}, v_i)} \bar{\rho_j} \simeq M$. It is possible that two consecutive nodes j and $j+1$ on the path satisfy the approximation requirement. Then v_i is to be chosen out of the two nodes. In the following we present three algorithms to make such a selection.

Algorithm LtoR: The first algorithm, called LtoR, computes the success probability of each segment from link 0 to link N-1 (left to right). If the success probability of a segment from link i to link j $(j > i)$ is closer to M than that of the segment from link i to link $j+1$, a converter is placed at node j. Otherwise, a converter is placed at node $j+1$.

The following function, Get_next_placement() as shown in Table 1, computes the location for the next converter and the success probability for the current segment, Y', given that the last converter location *last* and the geometric mean M. We use the same function for all the three algorithms. The value of Y' may or may not be used by some of the algorithms.

Table 1. Function: Get_next_placement()

```
int Get_next_placement(int last, float M, float *Y')
var float Y''; int i;
begin
   *Y' = ρ̄_last;
   for( i = last+1; i < N; i++) begin
     Y'' = *Y' ρ̄_i;
     if(fabs(*Y' − M) < fabs(Y'' − M) )
       return i;
     else
       *Y' = Y''
   end;
end.
```

The placement procedure continues until the last converter is placed or the end of the path is reached. A detailed description of this algorithm, LtoR, is given in Table 2.

Table 2. from Left to Right

```
LtoR(ρ̄_i, N, K)
var int np = i = 0; float M = (K+1)-th root of ∏_{j=0}^{N-1} ρ̄_j, Y' = 0;
begin
   while( (i < K+1) & (np < N) ) begin
     /*Compute the next converter placement
      given the last placement is at node np. */
      np ← Get_next_placement(np, M, &Y');
      Place a converter at np;
      i++;
   end;
end.
```

where $M = \sqrt[K+1]{\prod_{j=0}^{N-1} \bar{\rho}_j}$.

Algorithm $LM_{recomputed}$: The problem of using the algorithm, LtoR, is that all the approximation errors in Y_i for different segmentation may accumulate. Thus the last segment may have a large variation from the desired value of M. To reduce the effect of the approximation errors, the second algorithm, called $LM_{recomputed}$, is introduced. In this algorithm, instead of using the same value of M computed at the beginning of the algorithm, we recompute M every time a converter is placed. The success probability in a segment is compared with the new M. This algorithm is described in Table 3.

Algorithm LorR: Both of the above algorithms compute the success probability Y_i and compare it with M from the left nodes to the right nodes on a path. The same method can also be applied by computing Y_i and comparing it with M from the right nodes to left nodes on the path. To reduce the approximation errors and make the best selection so that each segment has the success probability as close as possible, we combine the above ideas together into algorithm three, LorR. In this algorithm, we compute the success probabilities of segments from both side of a path. A segment, which has success probability closer to M is selected regardless of being on the left or on the right, and a converter is placed at the node of that location. By viewing

Table 3. from Left to Right: recompute M

```
LM_recomputed(float ρ_i, int N, int K)
var int np = i = 0; float M = (K+1)-th root of ∏_{j=0}^{N-1} ρ̄_j, Y' = 0;
begin
   while((i < K+1) & (np < N)) begin
      /*Compute the next converter placement
      given the last placement is at node np. */
      np ← Get_next_placement(np, M, &Y');
      Place a converter at np;
      /*recompute M*/
      M = (K−i)-th root of (M^{K+1−i} / Y');
      i++;
   end;
end.
```

where $M = \sqrt[K+1]{\prod_{j=0}^{N-1} \bar{\rho}_j}$ and the recomputation is $M = \sqrt[K-i]{\frac{M^{K+1-i}}{Y'}}$.

the unselected links as a new path, M is recomputed. This algorithm is shown in Table 4.

Table 4. from Left or Right: recompute M

1. Compute M of the path as $M = \sqrt[K+1]{\prod_{j=0}^{N-1} \bar{\rho}_j}$;
2. Compute f'_L and f'_R from left to right and from right to left, respectively, using the function Get_next_placement().
3. Set $Y' = \min(\text{fabs}(f'_L\ \text{-M}), \text{fabs}(f'_R\ \text{-M}))$, and place a converter on the corresponding node;
5. If there are links not being considered, the links are viewed as a new path; recompute M as $M = \sqrt[k']{\frac{M^{k'+1}}{Y'}}$, where k' is the number of converters have not been placed; goto step 2;
   Else, stop.

Note that the actual locations of converters may vary but the overall blocking probability may still be the same if each segment has equal success probability on a path. To see this, consider a path with four nodes, n_0, n_1, n_2, n_3, and three links, l_0, l_1, l_2, with link loads $\rho_0 = \rho_1 = \rho_2$. A single converter can be placed at the node n_1 or n_2 to obtain the same blocking probability.

In the next section we show the results of using the above three algorithms, and compare them with the optimal solutions under different wavelength utilization on each link. Note that the first algorithm is the simplest and the third is the most complex among the three algorithms. Each of these three algorithms has linear complexity.

2.3. Numerical results and discussions

Table 5. Wavelength utilization on a 10-hop path

links	l_1	l_2	l_3	l_4	l_5
linear(**l**)	0.050	0.056	0.061	0.067	0.072
m-linear(**ml**)	0.100	0.125	0.150	0.175	0.200
exponential(**e**)	0.086	0.150	0.024	0.083	0.037
m-exponential(**me**)	0.006	0.012	0.024	0.048	0.096
uniform(**u**)	0.221	0.211	0.239	0.221	0.234

links	l_6	l_7	l_8	l_9	l_10
linear(**l**)	0.078	0.083	0.089	0.094	0.100
m-linear(**ml**)	0.200	0.175	0.150	0.125	0.100
exponential(**e**)	0.261	0.257	0.044	0.178	0.101
m-exponential(**me**)	0.096	0.048	0.024	0.012	0.006
uniform(**u**)	0.203	0.204	0.232	0.208	0.218

The three algorithms discussed above are evaluated in this section to find the optimal placement on a 10-hop path (N=10) as shown in Figure 1. The number of wavelength (F) on each link is also assumed to be 10. Five different wavelength utilization patterns are used as shown in Table 5. Linear (**l**) utilization in the table represents the wavelength utilization increasing linearly from 0.05 on link 0 to 0.1 on link N-1. The utilization of m-linear (**ml**) increases from link 0 to the middle of the path and then decreases to link N-1 with different rate. The utilization pattern **e** represents that the wavelength utilization are exponentially distributed on each link with mean 0.1 and variance 0.1. The wavelength utilization of m_exponential (**me**) increase with rate of 2 from link 0 to the middle of the path and then decreases with rate of $\frac{1}{2}$ to link N-1, and **u** represents the random utilization uniformly distributed between 0.2 and 0.3 on each link.

Table 6. Converter placements and blocking probability from different algorithms compared to optimal solution

	l	*ml*	*e*	*me*	*u*
Alg. 1	3 5 7 9	3 5 7 9	3 6 7 9	4 5 6 8	2 4 6 8
	4.61e-08	5.81e-05	2.80e-5	2.44e-10	4.26e-04
Alg. 2	3 5 7 9	3 5 6 8	3 6 7 9	4 5 6 7	2 4 6 8
	4.61e-08	4.22e-05	2.80e-5	2.39e-10	4.26e-04
Alg. 3	3 5 7 9	3 5 6 8	2 5 6 8	3 4 5 6	2 4 6 8
	4.61e-08	4.22e-05	7.49e-6	2.39e-10	4.26e-04
Opt.	3 5 7 9	3 5 6 8	2 5 6 7	2 5 6 7	2 4 6 8
	4.61e-08	4.22e-05	7.49e-6	2.39e-10	4.26e-04

The results of using the three algorithms are shown in Table 6 with different utilization patterns. The dynamic programming method guarantees to yield the optimal results that are shown in the table for comparision. It is noticed that all the algorithms yield optimal solutions for $K = 1, 2, 3$ on the 10-hop path. The results of placing $K = 4$ converters are shown in the table. Since we consider sparse wavelength converter networks, $K > 4$ is not of interest. We see from the table that optimal solutions are obtained under most of the utilization patterns. Some of the placements of using the segmentation algorithms are different from the dynamic programming results (i.e., algorithm 2, algorithm 3 and the optimal algorithm give three different solutions under the m_exponential utilization). However, all of them are optimal since the blocking probabilities are equal. Recall that the optimal result may not be unique. Among the three segmentation algorithms, the third algorithm is the most accurate. As mentioned earlier, although this algorithm is a little more complex than the others, it still has linear complexity. Many other utilization patterns are also tested and they support our observations. The results are not shown here because of the space limitation.

We noticed that an optimal result might not be obtained using the segmentation algorithm when the link loads are dramatically different; that is, some link loads are extremely high and other loads are extremely low. Under such utilization patterns, some links which have extremely light utilization, have little weight to the success probability ($\bar{\rho_i} > 0.9$) of a segment. Then the segmentation algorithms may incorrectly put such links in a segment. However, the blocking probability is affected very little by the lightly loaded links.

3. CONVERTER PLACEMENT ON A PATH CONSIDERING LINK-LOAD CORRELATION

In the previous section we considered the optimal converter placements to optimize the performance of the end-to-end calls. Two basic assumptions are made in order to deploy the binomial model: the call requests arrive at different wavelengths are statistically independent, and the link loads on a path are independent of each other. The wavelength independence is assumed to make the analysis simple. However, the second assumption may not be appropriate if the interfering traffic [3] arrives at each node, as is the case in a bus, a ring, or a path in an arbitrary topology network. A performance model considering link-load correlation was proposed in [3, 7]. In this section, we show that our segmentation algorithm is also applicable when the link-load correlation is considered, i.e., we assume that link loads are dependent.

3.1. Segmentation method using the link-load correlation model

The assumption of the previous section, i.e., the success probabilities in disjoint segments are statistically independent, is still needed when link-load correlation is considered. With this assumption, the success probability of an end-to-end call, $S(V)$, can still be computed as

$$S(V) = \prod_{i=1}^{K+1} f_i \tag{7}$$

where f_i is the success probability on segment i.

For lack of space, we omit explaining of the details of the link-load correlation model and ask the reader to refer to [3, 7] when necessary. Referring to Figure 1, we denote $P_n(i)$ be the probability that a new call enters the network at node i and uses link i on wavelength λ_1 given that λ_1 is not used by another call on link i. Let F be the number of wavelengths per fiber. Then the success probability on segment i, $f_i, (i \leq K+1)$, is given by

$$f_i = 1 - [1 - \bar{\rho}(v_{i-1}) \prod_{j \in L(v_{i-1}+1, v_i)} \bar{P}_n(k)]^F \ , \tag{8}$$

where $\bar{\rho}(i) = 1 - \rho(i)$ and $\bar{P}_n(i) = 1 - P_n(i)$.

Let Y_i be the success probability on one wavelength on segment i. Thus,

$$Y_i = \bar{\rho}(v_{i-1}) \prod_{j \in L(v_{i-1}+1, v_i)} \overline{P}_n(j) \ . \tag{9}$$

Our goal is to select a placement vector V to maximize $S(V)$ with

$$f_i = 1 - (1 - Y_i)^F \ . \tag{10}$$

We cannot apply Theorem 1 directly here because the product of Y_i is not a constant any more. Instead of having a set of positive numbers as in the link-load independence model, we have two sets of positive numbers, $\bar{\rho}(i)$ and $\bar{P}_n(i)$. For each segment, we need to select $\bar{\rho}(i)$ for the first link and $\bar{P}_n(i)$ for other links in each segment so that the product in each segment is approximately equal.

Note again that we are considering a sparse wavelength conversion network. The nodes, which have converters, are a small fraction of all nodes in the network. The success probability on segment i, Y_i defined in Eq. 9, usually include one $\bar{\rho}(i)$ and several $\bar{P}_n(i)$s. For a reasonable blocking probability of end-to-end calls, $\bar{\rho}(i)$, is not small. Therefore Y_i would be dominated by the product of $\bar{P}_n(i)$. Knowing these facts, we propose to use the following approximation:

Let X be the geometric mean of the ratios of $\bar{\rho}(i)$ to $\bar{P}_n(i)$; that is,

$$X = \sqrt[N]{\prod_{i=0}^{i=N-1} \frac{\bar{\rho}_i}{\bar{P}_n(i)}} \ .$$

X is a constant since both $\bar{\rho}_i$ and $\bar{P}_n(i)$ are known. Then $\bar{\rho}(i)$ can be approximately computed as

$$\bar{\rho}(i) \simeq \bar{P}_n(i)X$$

The product of $Y_i = \bar{\rho}(v_{i-1})\prod_{j\in L(v_{i-1}+1,v_i})\overline{P}_n(j)$ for all the segments is given as

$$\begin{aligned}\prod_{i=1}^{K+1} Y_i &= \prod_{i=1}^{K+1}(\bar{\rho}(v_{i-1})\prod_{j\in L(v_{i-1}+1,v_i)}\bar{P}_n(j)) \\ &\simeq \prod_{i=1}^{K+1}(X\bar{P}_n(v_{i-1})\prod_{j\in L(v_{i-1}+1,v_i)}\bar{P}_n(j)) \\ &= X^{K+1}\prod_{i=0}^{N-1}\bar{P}_n(i)\ . \qquad (11)\end{aligned}$$

Since X is a constant and the product of $\bar{P}_n(i)$ is also a constant on a path, the product of Y_i becomes a constant too, and $0 < Y_i < 1$. This approximation is shown to be valid when we compare the results obtained using this approximation with the optimal results in the next section. Note from Eqs 7, 10 and 11 that Theorem 1 is applicable to this problem because $\prod_{i=1}^{i=K+1} Y_i$ in Eq. 11 is a constant. Thus an optimal placement with the consideration of link-load correlation can be achieved if the success probability of each segment is equal on the path.

Following the previous section, to identify a segment, we can compute the success probability on one wavelength of successive links and compare it with the geometric mean of the success probability of each segment. The geometric mean of Y_i, M, can be derived as

$$M = X\ \sqrt[K+1]{\prod_{i=0}^{i=N-1}\bar{P}_n(i)}\ . \qquad (12)$$

By computing Y_i for each segment and comparing it with the target constant M, a path can be easily divided into $K+1$ segment. The first and second algorithms described in the previous section can also be applied here after a little modification: Eq. (12) is used instead of Eq. (6) to compute the geometric mean of the success probability on a segment. The function Get_next_placement(), used to compute the next converter location given the last converter is placed at node *last*, is replaced by a new function, Get_next_LtoR() shown in Table 7. In the new function, we compute the success probability using both ρ_i and $\bar{P}_n(i)$.

Table 7. Get next converter location from left to right considering link-load correlation

```
int Get_next_LtoR(int last, float M, float& Y')
var float Y''; int i;
begin
   Y' = ρ̄_last;
   for( i = last+1; i < N; i++) begin
     Y'' = Y' P̄_n(i);
     if(fabs(Y' − M) < fabs(Y'' − M) )
       return i;
     else
       Y' = Y''
     end;
end.
```

The third algorithm, selecting a segment from both left side and right side, cannot be applied directly because the first element of a segment success probability Y_i is $\bar{\rho}(i)$. It is difficult to predict where the first element is if we compute Y_i from right to left. However, this problem can be solved by considering the product of $\bar{\rho}_{i-1}$ and $\bar{P}_n(i)$ together instead of $\bar{\rho}_i$ or $\bar{P}_n(i)$ individually, when we compute a converter location from right to left. If the temporary success probability Y' is not close to M, the segment is expanded to the next link, and Y' is computed as $Y' = \frac{Y'}{\bar{\rho}_i}\ \bar{\rho}_{i-1}\bar{P}_n(i)$. A new version of the function Get_next_placement is given in Table 8.

Table 8. Get next converter location from right to left considering link-load correlation

```
int Get_next_RtoL(int last, floatM, float& Y')
var float Y''; int i;
begin
   Y' = ρ̄_last;
   for( i = last; i > 0; i−−) begin
     Y'' = (Y'/ρ̄_i) ρ̄_{i−1} P̄_n(i) ;
     if(fabs(Y' − M) < fabs(Y'' − M) )
       return i;
     else
       Y' = Y''
     end;
end.
```

The idea of selecting converter location from both sides can be used when link-load correlation is considered. A detailed description of the algorithm is shown in Table 9.

Table 9. from left or right considering link-load correlation

```
LorR_LLC(ρ̄, P̄_n, N, K)
var float f'_L=f'_R=0, Y' = 1, M = (K+1)√(X^K ∏_{i=0}^{i=N−1} P_n(i));
var int Lp=Rp = 0, path_begin = 0, path_end = N-1, i = 0;
begin     while( (i < K + 1) & (Lp < Rp)) begin
       M = √(M^{k'+1} / Y')
       Lp = Get_next_LtoR(path_begin, M, f'_L);
       Rp = Get_next_RtoL(path_end, M, f'_R);
       If ( fabs(f'_L - M) < f'_R - M)) begin
          /*f'_L is closer to M*/
          Place a converter at Lp;
          Y' = f'_L;
          path_begin = Lp;
          end;
       Else begin /*f'_R is closer to M*/
          Place a converter at Rp;
          Y' = f'_R;
          path_end = Rp;
          end;
       i++;
    end;
end.
```

3.2. Numerical results and discussions

The converter placement for different traffic matrices considering link-load correlation is compared with optimal results using dynamic programming in this section. Similar to the previous section, we set N = 10, and F=10. The link loads as shown in Table 10 are computed from the traffic matrix $[\lambda_{sd}]$. The constant link load is achieved by setting $\rho_n(0) = \rho$, and $\rho_n(i) = \rho - \sum_{j=0}^{i-1}\frac{H-i}{H-j}\rho_n(i-1), i =$

Table 10. Wavelength utilization computed from the traffic matrices

links	l_1	l_2	l_3	l_4	l_5
constant(**c**)	0.30	0.30	0.30	0.30	0.30
m-linear(**ml**)	0.10	0.12	0.24	0.28	0.30
nonuniform(**nu**)	0.12	0.19	0.24	0.27	0.28
uniform(**u**)	0.07	0.14	0.18	0.21	0.23
links	l_6	l_7	l_8	l_9	l_10
constant(**c**)	0.30	0.30	0.30	0.30	0.30
m-linear(**ml**)	0.30	0.28	0.24	0.12	0.10
nonuniform(**nu**)	0.28	0.27	0.24	0.19	0.12
uniform(**u**)	0.23	0.22	0.18	0.14	0.07

$1, 2, \ldots, N-1$, and $\lambda_{ij} = \frac{\rho_n(i)}{H-i}$ for $j > i$ [6]. The second utilization pattern, m-linear, is obtained by setting $\lambda_{ij} = 0.01, i, j = 0, 1, \ldots, N$. The traffic matrix of the third utilization pattern is set to $\lambda_{sd} = 0.04/|i-j|, i, j = 0, 1, \ldots, N$, which may be more practical in a path network. The last traffic matrix is randomly generated with uniform distribution U(0.005, 0.01).

Table 11. Converter placement and blocking probability with the consideration of the link load correlation

	nu_2	u_2	c_4	ml_4
Alg. 1	4 6	4 6	3 5 7 9	3 4 5 6
	2.88e-04	3.83e-06	1.71e-04	3.24e-05
Alg. 2	4 7	4 5	3 6 8 9	3 4 5 6
	3.52e-04	5.80e-06	1.63e-04	3.24e-05
Alg. 3	4 6	4 6	3 6 8 9	3 4 5 7
	2.88e-04	3.83e-06	1.63e-04	3.13e-05
Opt.	4 6	4 6	3 6 8 9	3 4 5 7
	2.88e-04	3.83e-06	1.63e-04	3.13e-05

The link loads are shown in Table 10. The corresponding converter locations and the blocking probabilities using the three algorithms are shown in Table 11. We observed from Table 11 that when the number of nodes with wavelength converters are small in a network and the link loads are not high for the reasonable blocking probability, the results obtained are same as that of using the dynamic programming method, which has the complexity of $O(N^2K)$.

A ring is a popular topology in optical networks. After placing a converter at a node in a ring network, the ring can be divided into a path as in [6]. Our results show that the algorithms are also applicable to ring networks. The results are omitted for lack of space.

4. CONCLUSIONS

In this paper, we considered the optimal converter placement problem for a given number of converters on a path topology. We first proposed and proved that optimal placement considering end-to-end calls are obtained when the segments on a path have equal success probability. The result of [6] that uniform placement of converters is optimal for the end-to-end performance when the link loads are uniform is a natural corollary included in this result. Then the theory was used to achieve optimal converter placement using both the link-load independence model and the link-load correlation model. Three implementation algorithms with linear complexity were introduced.

The results indicate that the optimal placement considering end-to-end calls can be obtained with linear complexity using the segmentation algorithm under different traffic patterns. Since a ring topology can be easily divided into a path [6], the algorithms can also be applied to ring networks.

APPENDIX

The following is a proof of lemma 1:

Proof: Let $\sum_{i=1}^{K+1} z_i = C$, where C is a constant. Let $g(z_i) = 1 - (1 - e^{z_i})^F$. We show below that $z_i^{opt} = C/(K+1), i = 1, 2, \ldots, K+1$ is an optimal vector to maximize $G(Z)$. To prove this, we need to show that

$$(K+1) \ln g(z_i^{opt}) \geq \sum_{i=1}^{K+1} \ln g(z_i)$$

where z_i^{opt} is an element of the optimal vector and z_i's are the elements of a feasible vector. Since z_i is in a convex set, we need to show that $\ln g(z)$ is a concave function of the continuous variable $z \in [-\infty, 0]$. Let $H(z) = \ln g(z)$. The second derivative of $H(z)$ is

$$H''(z) = \frac{g''(z)g(z) - [g'(z)]^2}{g^2(z)}.$$

Evaluating $g(z)$, $g'(z)$, and $g''(z)$ and substituting in the above equation, we obtain

$$H''(z) = \frac{Fe^z(1-e^z)^{F-2}(1 - Fe^z - (1-e^z)^F)}{g^2(z)}.$$

Note that for every $-\infty \leq z \leq 0$ and $F \geq 1$, $(Fe^z + (1-e^z)^F)$ is a non-decreasing function of z and is lower bounded by 1. Since the remaining factors are all positive, $H''(z) \leq 0$. ■

REFERENCES

[1] K. C. Lee and O. K. Li, "A wavelength-convertible optical entwork", *IEEE Journal on Lightwave Technology*, vol. 11, pp. 962-970, May/June, 1993.

[2] M. Kovačević and A. S. Acampora. Benefits of wavelength translation in all-optical clear-channel networks. *IEEE Journal on Selected Areas in Communications*, 14(5):868–880, June 1996.

[3] R. A. Barry and P. A. Humblet, "Models of blocking probability in all-optical networks with and without wavelength changers", *IEEE Journal on Selected Areas in Communications*, 14(5):858–867, June 1996.

[4] S. Subramaniam, M. Azizoğlu, and A. K. Somani. "All-optical networks with sparse wavelength conversion", *IEEE/ACM Transactions on Networking*, 4(4):544–557, Aug. 1996.

[5] J. Yates, J. Lacey, D. Everitt, and M. Summerfield. "Limited-range wavelength translation in all-optical networks", In *Proc. INFOCOM '96*, pages 954–961, March 1996.

[6] Subramaniam, M. Azizoglu, and A. K. Somani, "On the optimal placement of wavelength converters in wavelength-routed networks", In *Proc. INFOCOM'98*, pp. 902-909, April, 1998.

[7] R. A. Barry and D. Marquis, "Evaluation of a model of blocking probability in all-otical mesh networks without wavelegnth changers", *Proc. SPIE*, pp. 154-167, Oct. 1995.